Hindawi Publishing Corporation
Journal of Sensors
Volume 2016, Article ID 3184642, 11 pages
http://dx.doi.org/10.1155/2016/3184642

Research Article

RLT Code Based Handshake-Free Reliable MAC Protocol for Underwater Sensor Networks

Xiujuan Du,[1] Keqin Li,[2] Xiuxiu Liu,[1] and Yishan Su[3]

[1] *School of Computer Science, Qinghai Normal University, Xining, Qinghai 810008, China*
[2] *Department of Computer Science, State University of New York, New Paltz, NY 12561, USA*
[3] *School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China*

Correspondence should be addressed to Xiujuan Du; 124111397@qq.com

Received 11 September 2015; Accepted 4 January 2016

Academic Editor: Jian-Nong Cao

The characteristics of underwater acoustic channels such as long propagation delay and low bit rate cause the medium access control (MAC) protocols designed for radio channels to either be inapplicable or have low efficiency for underwater sensor networks (UWSNs). Meanwhile, due to high bit error, conventional end-to-end reliable transfer solutions bring about too many retransmissions and are inefficient in UWSN. In this paper, we present a recursive LT (RLT) code. With small degree distribution and recursive encoding, RLT achieves reliable transmission hop-by-hop while reducing the complexity of encoding and decoding in UWSN. We further propose an RLT code based handshake-free (RCHF) reliable MAC protocol. In RCHF protocol, each node maintains a neighbor table including the field of state, and packages are forwarded according to the state of a receiver, which can avoid collisions of sending-receiving and overhearing. The transmission-avoidance time in RCHF decreases data-ACK collision dramatically. Without RTS/CTS handshaking, the RCHF protocol improves channel utilization while achieving reliable transmission. Simulation results show that, compared with the existing reliable data transport approaches for underwater networks, RCHF can improve network throughput while decreasing end-to-end overhead.

1. Introduction

Recently, research on underwater sensor networks (UWSNs) has attracted significant attention [1–6] due to its potential application in environmental monitoring, resource investigation, disaster prevention, and so on. UWSN adopts acoustic communication, and acoustic channel is characterized by high bit error with the order of magnitude of 10^{-3}–10^{-7}, long propagation delay of a few seconds, and narrow bandwidth of kbps, resulting in terrestrial-based communication protocols being either inapplicable or inefficient for UWSN. Compared with conventional modems, acoustic modems in UWSN are more energy-consuming. However, nodes are battery-powered and harder to recharge and replace in harsh underwater environments. Furthermore, underwater nodes are usually deployed more sparsely; most nodes can move passively with water currents or other underwater activities, and some nodes will fail as energy depletion or hardware fault, so the network topology of UWSN usually changes dynamically, which brings about significant challenges to protocol design for UWSN.

The characteristics of acoustic communication discussed above make terrestrial protocols inapplicable in UWSN, regardless of MAC mechanism or reliable data transmission. MAC protocols have great impact on network system and are especially important for low quality channel. MAC protocols are usually divided roughly into two categories: contention based protocols and contention-free protocols. Contention based protocols are further divided into two subcategories: random access and collision-avoidance. In random access mechanism, a sender transmits packets without any coordination, which could lead to collision. In collision-avoidance protocols, RTS/CTS handshake is used widely to manage contention at both sides of sender and receiver, which can solve the hidden terminal problem and avoid collision between data packets to some extent. So collision-avoidance protocols outperform random access approaches in networks with heavy traffic. However, considering the characteristics of long propagation delay and low bandwidth of underwater acoustic channel, RTS/CTS handshake decreases the utilization of channel and is inefficient in UWSN.

Contention-free protocols are divided into three categories: TDMA, FDMA, and CDMA, in which channels are separated by time, frequency, or code domain, respectively. FDMA is unsuitable for UWSN because of the narrow available frequency range in underwater acoustic channels. TDMA requires time synchronization which is very challenging due to the variable delay in UWSN, and TDMA shows limited bandwidth efficiency because of the long time guards required in acoustic channel. CDMA separates signals by spreading codes and supports concurrent transmissions, which is resilient to multipath and Doppler's effects prevalent in underwater environments. Nevertheless, CDMA system requires additional hardware.

The characteristics of acoustic communication make conventional reliable transport mechanism inapplicable as well for UWSN, which is analyzed as follows.

(1) High bit error rate of acoustic channel leads to high erasure probability of packet and low success probability of transferring hop-by-hop. So, traditional end-to-end reliable transport mechanism may incur in too many retransmissions and collisions and reduce channel utilization.

(2) Low propagation speed of acoustic signals leads to long end-to-end delay, which brings about some issues for two end-nodes to control transmission timely.

(3) ARQ (Automatic Repeat Request) mechanism requires ACKs (acknowledgements) for the packets received successfully and retransmits the lost packets. It is well known that the channel utilization of simple stop-and-wait ARQ protocol is very low in UWSN with long propagation delay and low bit rate. In addition, acoustic modems adopt half-duplex communication, which limits the choice of efficient pipelined ARQ protocols. Even worse, if ACKs are lost, the packets received successfully will be retransmitted by the sender, and more bandwidth and energy will be consumed.

(4) Some reliable data transport protocols resort to FEC (Forward-Error-Correcting) to overcome the problems caused by ACKs. FEC adopts erasure codes and the amount of redundancy is fixed prior to transmission. Before transmitting, the sender encodes a set of n original packets into a set of N ($N \geq n$) encoded packets. Let $m = N - n$, so m redundant packets are generated. In order to reconstruct n original packets, the receiver has to receive a certain number (larger than n) of encoded packets. The stretch factor is defined as N/n, which is a constant that depends on erasure probability. However, the error probability of channel in UWSN is dynamic, and overestimated error probability will incur in additional overhead, whereas low-estimated error probability will lead to transmission failure.

The main contributions of this paper are summarized as follows.

(1) Underwater acoustic channels are characterized by high bit error and temporal and spatial variation. Based on digital fountain code, we propose a recursive LT (RLT) code with small degree distribution and introduce the erasure probability of channel p_p into RLT code for the first time, which improve the probability of decoding at the receiving node. RLT is applicable to dynamic UWSN with limit transmission time between two nodes. RLT reduces the overhead of encoding and decoding and improves the efficiency of decoding process substantially.

(2) Based on RLT, we provide a reliable and handshake-free MAC protocol which is called RCHF MAC protocol. In RCHF protocol, a sender transmits packets according to the state of the receiver, which can avoid collisions caused by transmitting to a node in sending state as well as transmitting to the same node by different senders at the same time. A minimum time interval T_a is set between two successive transmission phases of a node, and T_a is also called transmission-avoidance time, which could decrease collision between data packets and ACK packets. So, without RTS/CTS handshake, RCHF improves channel utilization and realizes reliable transmission hop-by-hop.

(3) To the best of our knowledge, SDRT [7] proposed by Xie et al. and CCRDT [8] proposed by Mo et al. are two reliable data transport protocols for UWSN up to date. We conduct a large number of simulation experiments to evaluate the performance of RCHF protocol and compare it with SDRT and CCRDT in aspects of throughput, overhead, end-to-end delay, delivery ratio, and energy-consumption. Simulation results show that RCHF has noticeable performance improvement.

The remainder of the paper is organized as follows. Section 2 briefly discusses related work. In Section 3, we describe RLT (recursive LT) code. Section 4 presents state based and handshake-free MAC protocol which achieves reliable data transmission hop-by-hop. Section 5 evaluates the performance of RCHF protocol through simulations. Finally, Section 6 concludes the paper and discusses some future work.

2. Related Work

2.1. Reliable Transmission Mechanism. Reliable data transmission over network has been the subject of much research. TCP ensures reliable delivery essentially through retransmitting the packets which have not been acknowledged by receivers, having poor performance in underwater acoustic channel impaired heavily. So, reliable transmission solutions based on coding were proposed [9, 10].

Reed and Solomon proposed Reed-Solomon code based on some practical erasure codes [11]. Reed-Solomon code is efficient for small values of n and m. However, the algorithms of encoding and decoding require field operations and result in too high computation overhead. Due to the limited computation capability of nodes, Reed-Solomon code is unsuitable for UWSN. Luby et al. studied a practical Tornado code [12]. Tornado code only involves XOR operations, and the algorithms of encoding and decoding are faster than Reed-Solomon code. However, Tornado code uses multilayer bipartite graph to encode and decode packets, resulting in high computation and communication overhead. Xie et al. presented SDRT reliable transport protocol. SDRT adopts SVT code to improve encoding/decoding efficiency. Nevertheless, after pumping the packets within the window quickly into the channel, the sender sends the packets outside the window at a very slow rate until receiving a positive feedback from the receiver, which reduces channel utilization. Mo et al. investigated a multihop coordinated protocol for UWSN based on GF(256) random-linear code to guarantee reliability and efficiency [8]. However, the encoding vectors are generated randomly, so the success probability of recovering K data packets from K encoded packets could not be guaranteed, and its decoding complexity is higher than other sparse codes. Furthermore, the multihop coordination mechanism requires time synchronization and is restricted in a string topology where there is a single sender and a single receiver.

Digital fountain codes are sparse codes on bipartite graphs with high performance [13, 14], which are rateless; that is, the amount of redundancy is not fixed prior to transmission and can be decided on the fly as the error recovery algorithm evolves. These codes are known to be asymptotically near-optimal for every erasure channel and allow for lightweight implementation of encoder and decoder. Luby put forward LT code, in which the decoder is capable of recovering the original symbols with high probability from any set of output symbols whose size is close to the origins [15]. However, LT code is designed for large number of data packets, which is not the case in UWSN, especially for mobile networks where transmission time between two nodes is very limited because of node mobility. Furthermore, the degree distribution used in LT code results in the large degree of nodes in the graph, which brings about large overhead for each packet.

2.2. MAC Protocol for UWSN. There are still many open issues in building underwater acoustic networks, and most of research work focused on the design of MAC protocols [16]. For example, Zhou et al. dealt with the design and performance evaluation of random access scheme in UWSN [17] and focused on tradeoffs arisen in clustered topology and proposed a protocol which is partly deterministic and partly random in [18]. Du et al. presented CDMA based MAC protocols for UWSN [19]. CDMA system requires additional hardware, and the narrow available frequency range in underwater acoustic channels has a negative impact on the application of CDMA based MAC protocols.

Most of the collision-avoidance MAC protocols in UWSN so far adopted RTS/CTS frames to coordinate transmission and ACK frames to acknowledge the data frames received successfully. However, RTS/CTS-based protocols are helpless in exposed terminal problem, and the channel utilization of handshake-based protocols is very low, especially in UWSN with long propagation delay and low bit rate. In addition, multiple handshakes considerably prolong end-to-end delay. So, RTS/CTS-based protocols are inefficient in impaired UWSN channel.

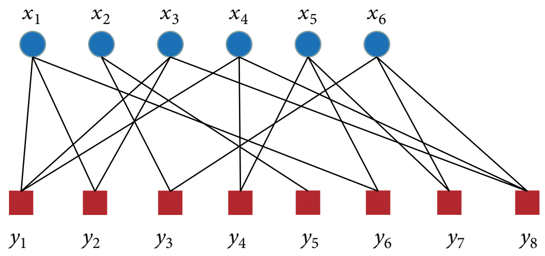


Figure 1: Encoding graph of LT code.

In this paper, a recursive LT (RLT) code with small degree distribution is proposed, which reduces the complexity of encoding and decoding. Integrated with RLT code, a reliable RCHF MAC protocol is presented. With high energy efficiency and channel utilization, RCHF MAC protocol achieves reliable transmission hop-by-hop for UWSN.

3. The RLT Code

RLT code is a variant of the LT code. Prior to RLT, the encoding and decoding of LT code are explained in this section.

3.1. The LT Code

3.1.1. Encoding. Consider a set of k input (original) packets, with each having the same length of l bits [20, 21]. The LT encoder takes k input packets and generates a potentially infinite sequence of encoded packets. Each encoded packet is computed independent of others. More precisely, given k input packets $\{x_1, x_2, \dots, x_k\}$ and a suitable probability distribution $\Omega(d)$, where d is the degree of encoded packets, the number of input packets used to generate the encoded packets, $d \in \{1, 2, \dots, k\}$, a sequence of encoded packets $\{y_1, y_2, \dots, y_j, \dots, y_n\}$, $n \geq k$, are generated as follows.

(1) Select randomly a degree d according to the distribution $\Omega(d)$.

(2) Select d input packets uniformly at random and set y_j equal to the bitwise sum modulo 2 of the d input packets. This can be implemented by successively XORing the d packets. y_j is an encoded packet with an index field used to indicate the IDs of the XORed input packets.

The relationship between the input and the encoded packets can be described by the graph in Figure 1 in which n encoded packets are generated from k input packets, where $n = 8$, $k = 6$, and the degree of y_1 is equal to three.

3.1.2. Decoding. The LT decoder tries to recover the original input packets from received encoded packets. The decoding process is as follows.

(1) Find an encoded packet y_j which is connected to only one input packet x_i and go to step (2). If the receiving node fails to find such encoded packet, stop decoding.

(2) Set $x_i = y_j$.

(3) Set $y_m = y_m \oplus x_i$ for each encoded packet which is connected to x_i, denoted by y_m. Here, $\oplus$ indicates XOR operation.

(4) Remove all edges connected to x_i.

(5) Go to step (1).

3.1.3. Soliton Distribution. For LT codes, the ideal degree distribution $\Omega(d)$ is the soliton distribution given by the following equation:

$$\Omega(d) = \begin{cases} \dfrac{1}{k}, & d = 1; \\ \dfrac{1}{d(d-1)}, & d = 2, 3, \ldots, k, \end{cases} \quad (1)$$

where $\sum_d \Omega(d) = 1$.

LT code is designed for the blocks including large number of data packets. However, it is not the case in mobile UWSN, where transmission time between two nodes is very limited due to node mobility. The degree distribution used in LT code incurs in large degree of nodes in the graph and brings about much overhead for encoding and decoding. The ideal soliton distribution of LT code shows perfect behavior in terms of the expected number of encoded packets needed to recover input packets. Unfortunately, like most ideal things, this distribution is quite fragile, in fact so much, so that it is useless in practice.

3.2. Recursive LT (RLT) Code. Coding scheme has great impact on system performance. In this section, we present recursive LT (RLT) code, achieving quickly encoding and decoding. For RLT code, we have assumptions that packet losses are independent. We use a bipartite graph $G = (V, E)$ with two levels to represent RLT code, where E is the set of edges and V is the set of nodes in the graph; $V = D \cup C$, where D is the set of input packets and C is the set of encoded packets. The edges connect the nodes in D and C.

3.2.1. Encoding. Given k input packets $x_1, x_2, \ldots, x_k \in D$ and degree distribution $\Omega(d)$, Let $n = (k + \xi)/(1 - p_p)$, where n is the expected number of encoded packets received successfully, p_p is erasure probability of underwater acoustic channel, that is, the packet error rate (PER), and ξ ($\xi > 0$) corresponds to the expected number of redundant encoded packets received. ξ redundant packets are used to decrease the probability that the receivers cannot restore the original k input packets through one transmission phase. The sequence of encoded packets is $y_1, y_2, \ldots, y_j, \ldots, y_n \in C$. The encoding procedure of RLT is as follows.

(1) From D, set input packets, successively XOR, the k packets to generate one encoded packet with degree k, and then duplicate the packet to get $\lceil 1/(1 - p_p) \rceil$ copies.

(2) From set D, select $\lceil m/(1 - p_p) \rceil$ distinct packets randomly to constitute a seed set S_1 and generate $\lceil m/(1 - p_p) \rceil$ encoded packets with degree one. Here, m is the expected number of encoded packets received successfully with degree one. In reality, we can set $1 \le m \le \max(\lfloor k/4 \rfloor, 1)$.

(3) Let $S_2 = D - S_1$. From the set S_2, select uniformly $\lceil k/(2(1 - p_p)) \rceil$ input packets at random, do XOR operation, respectively, with one packet in the set S_1selected randomly to generate $\lceil k/(2(1 - p_p)) \rceil$ encoded packets with degree two.

(4) Let $S_3 = D - S_1 - S_2$. If S_3 is not null, select $\lceil k/(6(1 - p_p)) \rceil$ input packets at random from the set S_3; otherwise, from the set D, do XOR operation, respectively, with one packet from S_2 and another from S_1 to generate $\lceil k/(6(1 - p_p)) \rceil$ encoded packets with degree three.

(5) Let $S_4 = D - S_1 - S_2 - S_3$. If S_4 is not null, select randomly $\lceil (\xi + k/3 - m - 1)/(1 - p_p) \rceil$ input packets from the set S_4; otherwise, from the set D, do XOR operation, respectively, with three packets from S_1, S_2, S_3, respectively, to generate $\lceil (\xi + k/3 - m - 1)/(1 - p_p) \rceil$ encoded packets with degree four.

3.2.2. Decoding. When an encoded packet is transmitted over an erasure channel, it is either received correctly or lost. The RLT decoder tries to recover the original input packets from the encoded packets received. The decoding process of RLT is as follows.

(1) Find an encoded packet y_j which is connected to only one input packet x_i. If the receiving node fails to find such encoded packet, stop decoding.

(2) Set $x_i = y_j$.

(3) Set $y_m = y_m \oplus x_i$ for each encoded packet which is connected to x_i, denoted by y_m. Here, $\oplus$ indicates XOR operation.

(4) Remove all edges connected to x_i.

(5) Go to step (1).

3.2.3. Degree Distribution. The limited delivering time between two nodes caused by node mobility leads to the fact that digital fountain codes are constrained to work with small k in UWSN communication. In order to reconstruct the input packets, the degree distribution of encoded packets received should have the following properties for RLT.

(1) The encoded packets received should involve all the input packets.

(2) The process of encoding and decoding should not involve too many XOR operations.

(3) At least one encoded packet with degree one should be received correctly by the receiver.

Given the high bit error, denoted by p_b, with the order of magnitude of 10^{-3}–10^{-7}, the PER p_p is given by the following equation:

$$p_p = 1 - \left(1 - p_b\right)^l, \tag{2}$$

where l is the packet size. Reference [22] gave the optimal packet size based on MICRO ANP protocol architecture, which is greater than one hundred bytes. So, given the optimal packet size, p_p is nontrivial. Considering k input packets, in order to address the aforementioned properties of degree distribution, the degree distribution of encoded packets in sending nodes is given by the following equation:

$$\Omega(d) = \begin{cases} \dfrac{m}{\xi + k}, & d = 1; \\ \dfrac{k}{d(d-1)(\xi + k)}, & d = 2, 3; \\ \dfrac{\xi + (1/3)k - (m+1)}{\xi + k}, & d = 4; \\ \dfrac{1}{\xi + k}, & d = k, \end{cases} \tag{3}$$

where $\sum_d \Omega(d) = 1$.

Lemma 1. *The average degree of encoded packets $\lambda \approx 3.7$.*

Proof. From the degree distribution given by (3), we get

$$\begin{aligned} \lambda = E(d) &= \sum_{d=1}^{4} \left(d \times \Omega(d)\right) \\ &= \frac{1 \times m}{\xi + k} + \frac{2 \times k}{2 \times 1 \times (\xi + k)} + \frac{3 \times k}{3 \times 2 \times (\xi + k)} \\ &\quad + \frac{4 \times (\xi + 1/3k - (m+1))}{\xi + k} + \frac{k}{\xi + k} \\ &= 3\frac{2}{3} + \frac{\xi/3 - 3m - 4}{\xi + k}. \end{aligned} \tag{4}$$

Usually, $|(\xi/3) - 3m - 4| \ll |\xi + k|$, so $\lambda \approx 3(2/3) \approx 3.7$. □

From Lemma 1, we can get that the decoding complexity of RLT is about 3.7 which is independent of the number of input packets. The comparison of encoding/decoding complexity is shown in Table 1.

3.3. Statistics Analysis of RLT Code.

Now, we define function $\psi_k(n)$ as the probability of decoding successfully when n encoded packets have been collected by the receiver. Note that $\psi_k(n) = 0$ for $n < k$. For completely random digital fountain code, $\psi_k(n)$ can be computed exactly as the probability of full rank of a $k \times n$ $(n \geq k)$ random binary matrix (i.e., a matrix with elements in GF(2)), which can be expressed as in the following equation:

$$\psi_k(n) = \prod_{i=0}^{k-1} \left(1 - 2^{(i-n)}\right). \tag{5}$$

Considering the erase probability p_p of channels, let $B(N; n, 1 - p_p)$ denote the probability of n packets received successfully, where N is the number of encoded packets transmitted by a sender, $N, n \in Z$, $n \leq N$, $0 \leq p_p \leq 1$, $0 \leq B(N; n, 1 - p_p) \leq 1$. $B(N; n, 1 - p_p)$ is the probability mass function (PMF) of binominal distribution and is given by the following equation:

$$B(N; n, p) = \binom{n}{N} p^n (1 - p)^{N-n}. \tag{6}$$

So, according to random LT code, when N packets are transmitted at a sender, the decoding probability $\varphi_{N,k}(p_p)$ at the receiver is given by the following equation:

$$\varphi_{N,k}\left(p_p\right) = \sum_{n=k}^{N} B\left(N; n, 1 - p_p\right) \left(\prod_{i=0}^{k} \left(1 - 2^{(i-n)}\right)\right). \tag{7}$$

However, according to the RLT with recursive encoding/decoding, the schematic of binary matrix of encoded packets received is as follows:

$$\begin{array}{r|c cccc cccc cccc cccc|}
\text{Degree:} & k & \multicolumn{4}{c}{1} & \multicolumn{4}{c}{2} & \multicolumn{4}{c}{3} & \multicolumn{4}{c}{4} \\
\text{Packet number:} & 1 & \multicolumn{4}{c}{\overbrace{\qquad\qquad}^{m}} & \multicolumn{4}{c}{\overbrace{\qquad\qquad}^{k/2}} & \multicolumn{4}{c}{\overbrace{\qquad\qquad}^{k/6}} & \multicolumn{4}{c}{\overbrace{\qquad\qquad}^{\xi - m - 1 + k/3}} \\
x_1 & 1 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 1 & 1 & 0 & \cdots & 1 \\
x_2 & 1 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 1 & 1 & 1 & \cdots & 0 & 1 & 1 & \cdots & 1 \\
x_3 & 1 & 1 & 0 & \cdots & 0 & 0 & 1 & \cdots & 1 & 0 & 1 & \cdots & 1 & 0 & 1 & \cdots & 1 \\
x_4 & 1 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 1 & 1 & 1 & \cdots & 1 \\
x_5 & 1 & 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 & 1 & 1 & \cdots & 0 \\
, & , & , & , & \cdots & , & , & , & \cdots & , & , & , & \cdots & , & , & , & \cdots & , \\
\vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\
, & , & , & , & \cdots & , & , & , & \cdots & , & , & , & \cdots & , & , & , & \cdots & , \\
x_k & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\hline
 & y_1 & y_2 & y_3 & \cdots & , & , & , & \cdots & y_j & , & , & \cdots & , & , & , & y_{\xi+k} & ,
\end{array} \tag{8}$$

Table 1: Decoding complexity comparison.

Codes	Encoding/decoding complexity
GF(256) in [8]	$O(k^3)$
LT	$k\ln_e^k$
SDRT in [7]	$k\cdot\ln(1/\varepsilon)$
RS	$k(N-k)\log_2^N$
RLT	3.7

In the binary matrix, the expected number of encoded packets received with degree k is “1”; the number with degree “1” is m; the number with degree “2” is $k/2$, and those packets are encoded fully recursively. In the encoding of RLT, S_3 and S_4 may overlap with the set of D, which results in the fact that the decoding probability at the receiver is less than “1”. After $k+\xi$ encoded packets have been collected, the probability $\psi_k(n)$ of decoding successfully is as follows:

$$\psi_k(n) = \psi_k(k+\xi) > \prod_{i=0}^{(\xi-m-1+k/2)} \left(1-2^{(i-k-\xi)}\right). \quad (9)$$

We define function $\varphi_{k,\xi}(p_p)$ as the probability of recovering k input packets at the receiver after $(\xi+k)/(1-p_p)$ packets are transmitted at a sender. This probability can be given by the following equation:

$$\begin{aligned}\varphi_{k,\xi}\left(p_p\right) &= \sum_{\xi=0}^{\xi} B\left(\frac{\xi+k}{1-p_p}; k+\xi, 1-p_p\right)\psi_k(k+\xi)\\ &> \prod_{i=0}^{(\xi-m-1+k/2)}\left(1-2^{(i-k-\xi)}\right).\end{aligned} \quad (10)$$

4. RLT Code Based Handshake-Free Reliable MAC Protocol

Once the problems of degree distribution, encoding, and decoding of RLT code are solved in advance, a reliable RLT-based media access control protocol is needed for nodes to communicate in real time. Wireless transceivers often work in half-duplex mode, and a sending node equipped with a single channel cannot receive packets at the same time [20], so RCHF solution should be able to avoid interference caused by transmitting to a node in sending state. So far, in MAC solutions of wireless multihop packet networks, RTS/CTS handshake is usually used to determine dynamically if the intended receiver is ready to receive a frame. For underwater sensors, the generating rate of data bits is about 1–5 bps and the optimal packet-load for UWSN is about one hundred of bytes [22]; whereas the length of RTS is a few dozen bytes. So, RTS/CTS frames are not too small compared with data frames, and the benefits from RTS/CTS handshake are unremarkable. On the contrary, considering the characteristics of acoustic communication such as low bandwidth, long propagation delay, RTS/CTS handshake decreases channel utilization and network throughput dramatically while prolonging end-to-end delay. So, coupled closely with RLT code, we put forward a state based, handshake-avoidance reliable MAC solution for UWSN which is detailed in this section.

4.1. Reliable Transmission Mechanism. In RCHF MAC solution, a source node first groups input packets into blocks of size k; that is, there are k input packets in the block. Then, the source node encodes the k packets and sends the encoded packets into network. A block size of 50 requires that the minimum time interval for the communication between a sender and a receiver is about 60 s [7], which meets the limited transmission time between the two nodes in UWSN. By setting the block size k appropriately, RCHF can control the transmission time and allow the receiver to be able to receive enough encoded packets in order to reconstruct the original block even when the nodes are moving. The encoded packets are forwarded from the source to the destination block by block and each block is forwarded hop-by-hop.

In RCHF protocol, the nodes which are sending packets are thought to be in transmission phase. In order to avoid the collision of transmission synchronization between data and ACK, reduce the overhead from overredundancy and compromise between transmission efficiency and fairness, two transmission constraints are defined as follows.

(1) The maximum number of data frames allowed to transmit in one transmission phase is $N_{\max}$.

(2) The minimum time interval between two transmission phases of the same node is T_a. The node waiting for T_a expiration is considered to be in transmission-avoidance phase. Present underwater acoustic modems are half-duplex, the delay of state transition between sending and receiving usually ranges from hundreds of milliseconds to several seconds, which is close to the magnitude of the maximum round-trip time (RTT) [18]. To facilitate to the receiver to switch to sending state to transmit the ACK, we set $T_a = 2 \times \text{RTT}$.

After transmitting N, $(N \leq N_{\max})$ encoded packets, the sender switches to the receiving state, waiting for the ACK from the receiver. In order to reconstruct the original k input packets with high probability at the receiver, the number of encoded packets received successfully should be larger than k, which is $k+\xi$. Considering the high packet error rate p_p, we set $N = (k+\xi)/(1-p_p)$. The parameter ξ, $(\xi > 0)$ is fixed and corresponds to the expected number of redundant encoded packets received by the receiver. ξ redundant packets are used to decrease the probability that the receiver fails to restore the original k input packets through one transmission phase and the factor $1/(1-p_p)$ is used to compensate for some channel errors.

In ACK, the number of frames received at the receiver are included, which can be used to update the packet error rate p_p on this hop on the fly, as well as the indices of input packets unrecovered. If the receiver can reconstruct the whole block, it sends back an ACK with “null” in the index field.

Given k_1 input packets unrecovered after the previous transmission phase, the sender encodes and transmits N_1 encoded packets, $N_1 = (k_1+\xi)/(1-p_p)$, with the degree

Table 2: The state table of neighbor nodes.

Value	State
0	Sending state
1	Receiving frame from other nodes
2	Unknown state
3	Send-avoidance

distribute given by (3) in RLT code, in which k is replaced by k_1 in the next transmission phase, and then collects feedback from the receiver repeatedly until receiving the ACK with “null” in the index field.

4.2. State Based Handshake-Free Media Access Control. After network initialization, each node maintains one neighbor table dynamically as in Table 2, which includes a state field recording the real-time state of neighbor nodes. In Table 2, state “0” indicates that the neighbor node is in sending state, state “1” indicates that the neighbor node is receiving frames from other nodes, “2” means unknown state, and “3” means the neighbor node is in send-avoidance phase. The format of frames in our protocol is in Table 3. The frame sequence number is used to identify the frame of one frame chain in one transmission phase, the field of original packet ID is used to indicate the IDs of packets which are XORed, and the immediate ACK field is used to inform whether the receiver should return a ACK immediately or not, in which “1” is for yes and “0” is for no.

As a node has packets to send, it searches the neighbor table for the state field of the intended receiver. If the state is “0” or “1,” it will put off delivery till the state is greater than one; otherwise, the sender will turn into transmission phase and start to deliver frames. The pseudocodes for sending data are described in Algorithm 1. In lines (4)–(9), we give the pseudocodes of encoding a block and encapsulating. In lines (11)–(13), we give the pseudocodes of waiting for delivery when the receiver is determined to be busy. In lines (14)–(41), we give the pseudocodes of delivering a block when the state of the receiver is unknown or send-avoidance. From lines (17)–(21), we can see that, only after the first frame is acknowledged by the receiver, the sender will transmit subsequently the other frames in the block.

The pseudocodes for collecting the state information of neighbor node are described in Algorithm 2.

5. Performance Evaluation

5.1. Simulation Result of RCHF MAC Protocol. In this section, we evaluate the performance of RCHF MAC protocol by simulation experiments. All simulations are performed using NS2 (Network Simulator 2) with an underwater sensor network simulation package extension (Aqua-Sim). Our simulation scenario is similar to the reality, and one hundred nodes are distributed randomly in an area of 7000 m × 7000 m × 2000 m. Simulation parameters are shown in Table 4.

The protocol is evaluated in terms of average end-to-end delay, end-to-end delivery ratio, energy-consumption as in Figure 2, and throughput as in Figure 3. We define the delivery ratio and throughput of RCHF protocol as follows.

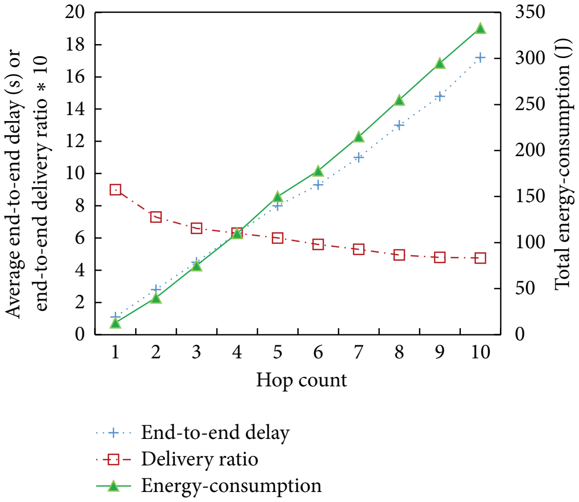


Figure 2: Performance versus hop count of RCHF.

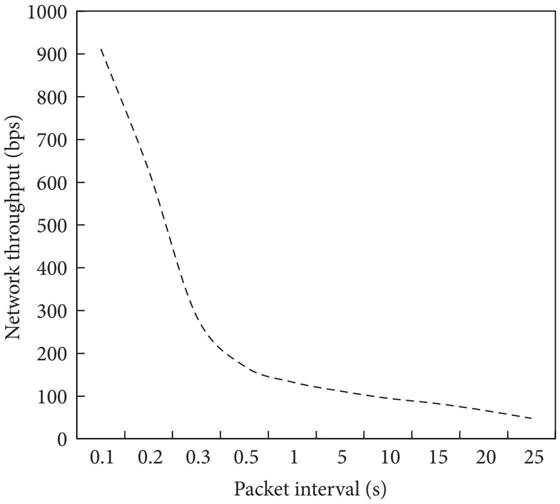


Figure 3: Network throughput versus packets interval of RCHF.

(1) The end-to-end delivery ratio is defined as the following equation:

$$\text{end-to-end delivery ratio} = \frac{\text{\# of packets received successfully at sink}}{\text{\# of packets generated at sources}}. \tag{11}$$

(2) The throughput is defined as the number of bits delivered to the sink node per second (bps).

From Figure 2, we can observe that the end-to-end delivery ratio of RCHF protocol is close to “1” when the hop count is “1” and decreases slightly with the increase of hop count, which is considered to be good performance for UWSN in the aspect of delivery ratio. Figure 2 also shows that the end-to-end delay and total energy-consumption rise with

```
(1)  INPUT: data block
(2)  OUTPUT: encoded packets delivery
(3)  Upon receiving a data block of size $k$ do
(4)    encodes $k$ packets to generate $N$ encoded packets;
(5)    set the frame sequence number of the first encoded packet to $N$;
(6)    // used to notice the overhearing nodes that the sender will transmit $N - 1$ frames subsequently
(7)    set the frame sequence number of the second packet to $N - 1$, and so on till last to “1”;
(8)    // also used to notice the overhearing nodes the frame number transmitted subsequently
(9)    set the “immediate ACK” field of the first frame as well as the last to “1”, others to “0”;
(10)   find the next hop (receiver) in routing table;
(11)   while (the state of the receiver ≤1) then
(12)     put off delivery;
(13)   endwhile
(14)    if (the state of the receiver >1) & (backoff_timer is timeout) then
(15)     switch to sending state and transmit the first frame; // start one transmission stage
(16)     switch to receiving state; // waiting for the ACK
(17)     if (receive the ACK for the first frame with “0” in the index field) then
(18)         switch to sending state and send the other frames subsequently;
(19)         set backoff_time = $T_a$;
(20)         switch to receiving state; // waiting for ACK for the block
(21)         // the first transmission stage for the block is over.
(22)         if (receive the ACK for the block) then
(23)           if ($k_1$ frames of the block are unrecovered successfully, ($k_1 \leq k$)) then
(24)              set $k = k_1$;
(25)              goto line (4);
(26)           endif
(27)       else (not receive the ACK for the block)
(28)              if (the state of the receiver >1) & (backoff_timer is timeout) then
(29)                 switch to sending state and send the last frames again;
(30)                 switch to receiving state; // waiting for ACK for the block again;
(31)              else
(32)                 goto line (28);
(33)              endif
(34)                 goto line (22);
(35)       endif
(36)     else (not receive the ACK for the first frame)
(37)       set backoff_time = $T_a$;
(38)       goto line (11);
(39)     endif
(40)    endif
(41) endupon
```

Algorithm 1: Encoding and transmitting.

```
(1)  INPUT: overhear a frame from a neighbor
(2)  OUTPUT: refresh the state of the neighbor’s in neighbor table
(3)  Upon overhearing a frame do
(4)    if (not an ACK) & (the sequence number >1) then
(5)      set the state of the neighbor to “0”;
(6)    else if (not an ACK) & (the sequence number = 1) then
(7)      set the state of the neighbor to “3”; // indicates the unknown state
(8)    else if (is an ACK with null index) then
(9)      set the state of the neighbor to “2”; // indicates the unknown state
(10)   else if (is an ACK) & (the value of index field is “0”) then
(11)     set the state of the neighbor to “1”; // ready to receive frames from other nodes
(12)   endif
(13) endupon
```

Algorithm 2: Collecting the state information of neighbor node.

Table 3: The format of data frame.

Bits: 8	8	8	2	6	1	23	Variable	16
Previous node layer	Previous node ID	Next node ID	Type 00: data 01: ACK	Frame sequence number	Immediate ACK	The IDs of Original packets	data	FCS
Head							load	tail

Table 4: Simulation parameters.

Parameter	Value
Block size k	50
Packet length l	160 bytes
Bandwidth	10 kbps
Routing protocol	static
Traffic	CBR
Transmission range	1500 m
MAC protocol	802.11

hop count, which is easily understood. Notice that the real value of end-to-end delivery ratio is the value of ordinate axis divided by 10.

As shown in Figure 3, the network throughput of RCHF decreases with the interval time between two successive packets generated by the source node. The reason is that the longer the interval time, the less the packets generated and the less the network load.

5.2. Performance Comparison. To the best of our knowledge, SDRT [7] and CCRDT [8] are two reliable data transport protocols for UWSN up to date. In this section, we conduct a large number of simulation experiments and compare the performance of RCHF with SDRT and CCRDT in the aspects of throughput and overhead. To facilitate comparing, the network parameters in this section are set as SDRT and CCRDT. The nodes are in a 4-hop string topology; the block size k is 5; the packet length l is between 50 and 200 bytes; the max bit rate is 800 bps. We use Poisson traffic generator to generate packets with time interval following a Poisson distribution. The overhead is defined as the ratio between the number of extra frames transmitted over multiple hops in the network and the number of original input packets.

The impact of packet length on the throughput of three protocols is shown in Figure 4(a). We can see that with different packet lengths from 50 to 200 bytes, RCHF achieves better end-to-end throughput than SDRT and CCRDT. Also, we observe that end-to-end throughput of three protocols goes up with the length of packets. However, the throughput of RCHF and CCRDT rise faster than SDRT with the growth of packet length. One of the reason is that RCHF and CCRDT adapt to the increasing PERs better than SDRT due to the PERs estimation, and RCHF can get more exact estimation of PERs by including the information of frames which are damaged or lost in a transmission phase in the ACK of RCHF.

The impact of bit rate on end-to-end throughput is shown in Figure 4(b), where packet length is 200 bytes. Again RCHF outperforms SDRT and CCRDT. The reason is that RLT code in RCHF adopts recursive encoding with higher efficiency than fully random encoding, so it reduces the amount of encoded frames necessary received to reconstruct the block.

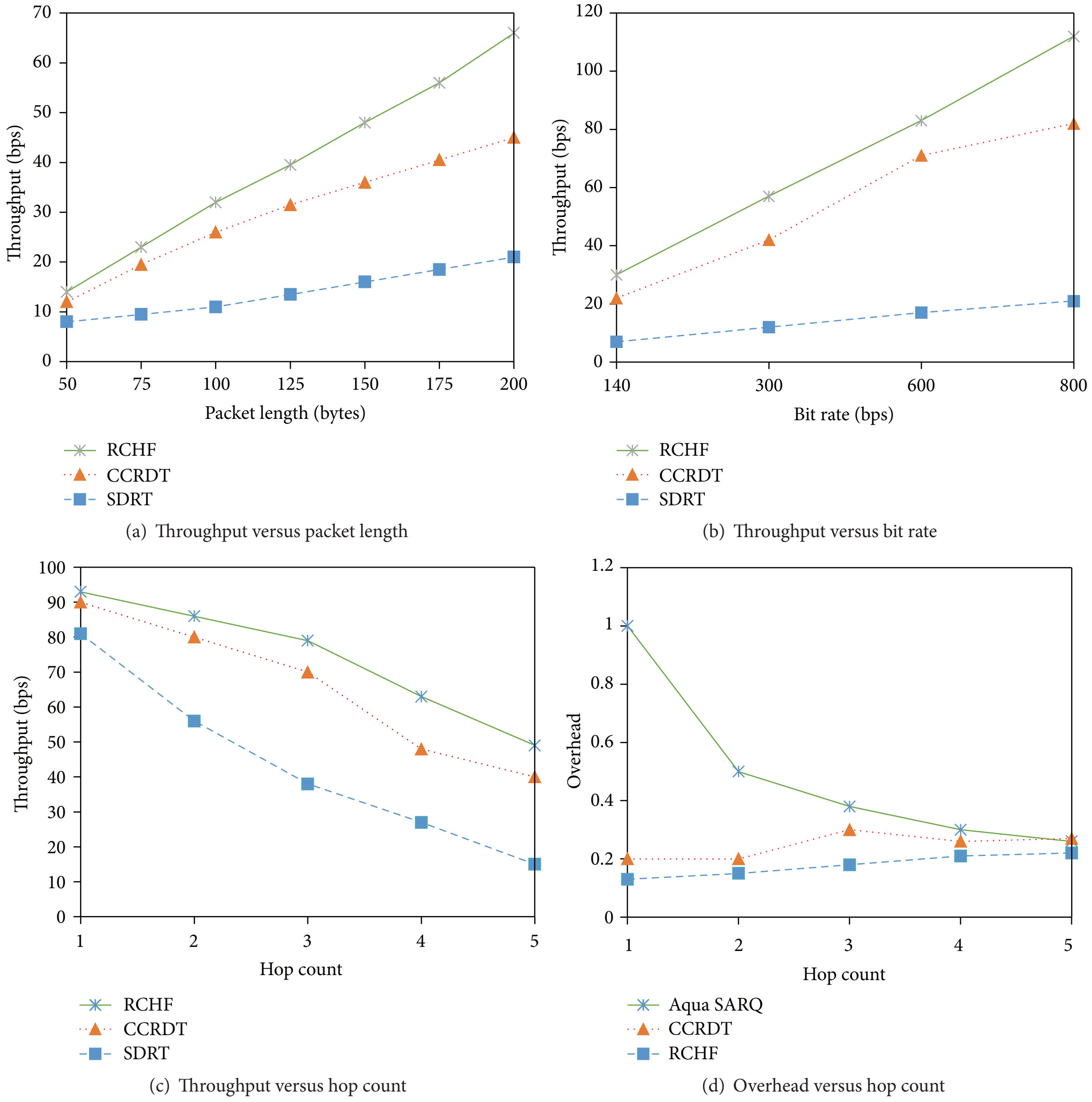
The impact of hop count on end-to-end throughput is shown in Figure 4(c). We can see that RCHF achieves the highest throughput among the three protocols. As the hop count rises, the throughput of all three protocols decreases, and the throughput of SDRT degrades faster than the other two protocols. That is because the packages in RCHF are forwarded according to the state of the receiver and can avoid the sending-receiving collisions and overhearing collisions; further, the time interval of transmission-avoidance in RCHF decreases dramatically the data-ACK collision, whereas SDRT has a larger chance to suffer from collisions with more hops. So, from Figure 4(c), we can see that RCHF MAC protocol improves channel utilization remarkably.

The impact of hop count on overhead is shown in Figure 4(d). As aforementioned, overhead is defined as the ratio between the number of extra frames transmitted over multiple hops in the network and the number of original input packets, where the number of extra frames is denoted by $\sum_{i=1}^{m}(N_i - k_i)$, and i denotes the ith transmission phase. We can see that the overhead of RCHF and CCRDT stays mostly unchanged for different hop counts primarily because the PERs estimation of RCHF and CCRDT minimizes the amount of unnecessary encoded packets. Also, we can see that RCHF achieves a smaller overhead than the other two protocols for its efficient recursive encoding.

6. Conclusions

In this paper, we propose a kind of digital fountain code, called recursive LT code (RLT). With small degree distribution and recursive encoding, RLT achieves reliable transmission for UWSN hop-by-hop while reducing the complexity of encoding and decoding. Integrated with reliable transmission mechanism based on RLT code, we present RCHF MAC protocol. In RCHF protocol, frames are forwarded according to the state of the receiver which can avoid the sending-receiving collisions and overhearing collisions. In order to reduce the overhead from overredundancy and compromise between transmission efficiency and fairness, two transmission constraints are defined. The time interval of transmission-avoidance in RCHF decreases dramatically the data-ACK collision. Simulations show that RCHF protocol can provide higher delivery ratio and throughput and lower end-to-end delay and resource consumption.

As future work, we plan to implement RCHF protocol on real UWSN nodes and conduct extensive experimental tests



(a) Throughput versus packet length

(b) Throughput versus bit rate

(c) Throughput versus hop count

(d) Overhead versus hop count

Figure 4: Performance comparison.

using real UWSN nodes in order to evaluate the performance and improve the design of RCHF protocol.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Authors' Contribution

The contributions of the author Xiujuan Du focused on the design of RCHF MAC protocol and he wrote the paper. Keqin Li put forward a lot of valuable advice as a director. Xiuxiu Liu and Yishan Su conducted many experiments.

Acknowledgments

This work is supported by the National Natural Science Foundation Projects of China (61162003, 61163050, and 61571318), Qinghai Office of Science and Technology (2015-ZJ-904, 2012-Z-902).

References

[1] K. Hao, Z. Jin, H. Shen, and Y. Wang, "An efficient and reliable geographic routing protocol based on partial network coding for underwater sensor networks," *Sensors*, vol. 15, no. 6, pp. 12720–12735, 2015.

[2] X.-J. Du, K.-J. Huang, S.-L. Lan, Z.-X. Feng, and F. Liu, "LB-AGR: level-based adaptive geo-routing for underwater

sensor network," *The Journal of China Universities of Posts and Telecommunications*, vol. 21, no. 1, pp. 54–59, 2014.

[3] F. Hong, Y.-L. Zhang, B.-Z. Yang, Y. Guo, and Z.-W. Guo, "Review on time synchronization techniques in underwater acoustic sensor networks," *Acta Electronica Sinica*, vol. 41, no. 5, pp. 960–965, 2013.

[4] P. Xie and J.-H. Cui, "An FEC-based reliable data transport protocol for underwater sensor networks," in *Proceedings of the 16th International Conference on Computer Communications and Networks (ICCCN '07)*, pp. 747–753, IEEE, Honolulu, Hawaii, USA, August 2007.

[5] P. Casari, M. Rossi, and M. Zorzi, "Fountain codes and their application to broadcasting in underwater networks: performance modeling and relevant tradeoffs," in *Proceedings of the 3rd ACM International Workshop on Underwater Networks (WUWNet '08)*, pp. 11–18, ACM, New York, NY, USA, September 2008.

[6] G. Anastasi, E. Borgia, M. Conti, and E. Gregori, "HI: an hybrid adaptive interleaved communication protocol for reliable data transfer in WSNs with mobile sinks," in *Proceedings of the 7th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '09)*, pp. 1–6, Galveston, Tex, USA, March 2009.

[7] P. Xie, Z. Zhou, Z. Peng, J.-H. Cui, and Z. Shi, "SDRT: a reliable data transport protocol for underwater sensor networks," *Ad Hoc Networks*, vol. 8, no. 7, pp. 708–722, 2010.

[8] H. Mo, Z. Peng, Z. Zhou, M. Zuba, Z. Jiang, and J.-H. Cui, "Coding based multi-hop coordinated reliable data transfer for underwater acoustic networks: design, implementation and tests," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '13)*, pp. 4566–4571, IEEE, Atlanta, Ga, USA, December 2013.

[9] H. Mo, Z. Zhou, M. Zuba, Z. Peng, J.-H. Cui, and Y. Shu, "Practical coding-based multi-hop reliable data transfer for underwater acoustic networks," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '12)*, pp. 5751–5756, Anaheim, Calif, USA, December 2012.

[10] S. Ahmad, R. Hamzaoui, and M. Akaidi, "Unequal error protection using LT codes and block duplication," in *Proceedings of the 9th Middle Eastern Simulation Multi Conference*, pp. 104–108, Ghent, Belgium, 2008.

[11] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, pp. 300–304, 1960.

[12] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Practical loss-resilient codes," in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC '97)*, pp. 150–159, El Paso, Tex, USA, May 1997.

[13] D. J. C. MacKay, "Fountain codes," *IEE Proceedings—Communications*, vol. 152, no. 6, pp. 1062–1068, 2005.

[14] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[15] M. Luby, "LT codes," in *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, vol. 12, pp. 271–280, Vancouver, Canada, November 2002.

[16] G. Fan, H. Chen, L. Xie, and K. Wang, "An improved CDMA-based MAC protocol for underwater acoustic wireless sensor networks," in *Proceedings of the 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '11)*, pp. 1–4, IEEE, Wuhan, China, September 2011.

[17] Z. Zhou, Z. Pengt, J.-H. Cui, and Z. Jiang, "Handling triple hidden terminal problems for multi-channel MAC in long-delay underwater sensor networks," in *Proceedings of the International Conference on Computer Communications (INFOCOM '10)*, pp. 1–5, IEEE, San Diego, Calif, USA, March 2010.

[18] Z. Zhou, H. Mo, Y. Zhu, Z. Peng, J. Huang, and J. Cui, "Fountain code based adaptive multi-hop reliable data transfer for underwater acoustic networks," UCONN CSE Technical Report UbiNet-TR10-02, 2010.

[19] X. Du, C. Peng, X. Liu, and Y. Liu, "Hierarchical code assignment algorithm and state-based CDMA protocol for UWSN," *China Communications*, vol. 12, no. 3, pp. 50–61, 2015.

[20] T. Plank, "A tutorial on reed-solomon coding for software-practice fault-tolerance in raid-like systems," *Practice & Experience*, vol. 27, no. 9, pp. 995–1012, 1997.

[21] S. Laendner and O. Milenkovic, "LDPC codes based on latin squares: cycle structure, stopping set, and trapping set analysis," *IEEE Transactions on Communications*, vol. 55, no. 2, pp. 303–312, 2007.

[22] X. Du, S. Lan, F. Liu, and Z. Feng, "Micro-ANP network protocol architecture and simulation implementation," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 11, no. 4, pp. 1757–1768, 2013.

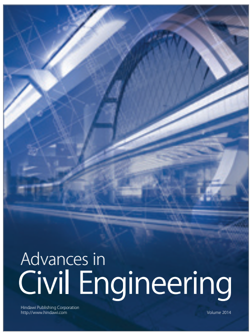
Journal of
Engineering

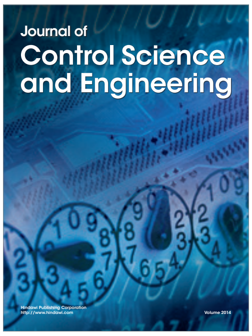
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

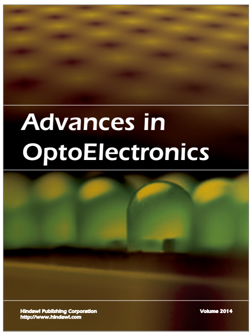
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

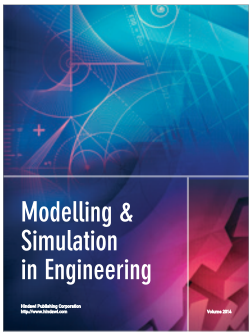
Journal of
Robotics

Journal of
Electrical and Computer Engineering

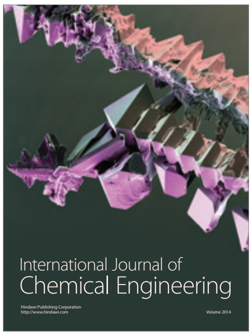

Advances in
OptoElectronics

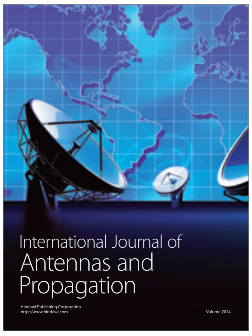
VLSI Design

International Journal of
Navigation and Observation

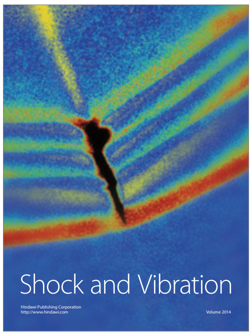

Modelling & Simulation in Engineering

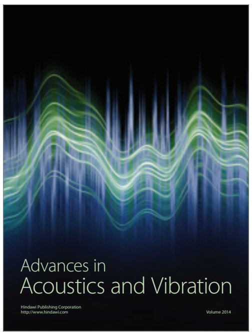
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration